

# **File Transfer Programs**

The GLAST ground system needs an automated system to transfer data between ground system elements. There has been a lot of discussion between various personnel in each ground element about available systems and their features and shortcomings. One problem with the discussion is that there does not seem to be an agreed upon set of requirements that must be met. In order to address this, we have made our cut at the requirements on this system (section 1). Hopefully this list can be refined to an agreed upon set.

Following the requirements, the next section discusses the findings of the trade study underway at the GSSC to evaluate FASTcopy and DTS.

## **1. Draft Requirements for a file transfer system.**

1. The program must be able to transfer arbitrary format files (binary or ASCII) from one ground element to another electronically across a network.
2. The program must be secure (no passwords sent in the clear, what else?).
3. The program must be able to verify that transferred files are intact – e.g., must verify a checksum or use some other method.
4. The system must be reliable
  - a. Must be able to recover gracefully from errors.
  - b. Must be able to detect transfer failures.
  - c. Must be able to re-try transmissions automatically.
5. Must be able to initiate execution of a command on the receiver end as soon as transfer has successfully completed (part of automation requirement).
6. Must be speedy enough able to transfer large amounts of data (up to 50 GB/day from the LAT ISOC to the GSSC – better than 5 Mb/s sustained.).
7. Must be able to resume file transfer from (near) where it left off after a failure instead of beginning again (this may be related to the previous requirement).
8. Must be able send out failure notification:
  - a. To sender on failure when unable to re-try transfer.
  - b. To receiver after multiple re-sends fail.
9. Must be able to record transactions on both sides of transfers, so both sides have the ability to automatically agree on what happened. (Data accountability requirement) .
10. The system must be able sense when not enough space is available for a file transfer to occur. (Automation requirement)
11. Must provide logging capability on both sides of transfer.

## **2. Trade Studies**

To understand what file transfer system we want to use, several ground elements have initiated evaluations of different pieces of software.

## 2.1.MOC Trade Study

The MOC evaluated 6 different types of file transfer solutions. Their clear favorite was Softlink's FASTcopy ([http://www.softlink.com/fast\\_non\\_techie.html](http://www.softlink.com/fast_non_techie.html)), because it was relatively cheap, provided needed functionality, and was quite flexible.

|  | Softlink   | Tumblewe<br>ed                     | Proginet   | Sterling                           | GNU   |   |
|--|--|------------------------------------|--|------------------------------------|---|---|
|  | <b>FASTCopy</b>  | <b>Secure<br/>Transport</b>        | <b>CyberFu<br/>sion</b>                          | <b>Connect<br/>Direct<br/>FTP+</b> | <b>rsync</b>  | <b>DTS</b>  |
| <b>Reliable</b>                                    | yes  | yes                                | yes  |                                    | yes*<br>retry not<br>automatic                                | no guarantee<br>mail delivery   |
| <b>Secure</b>                                      | SSL  | yes                                | yes  | yes                                | ssh   | yes (sftp)  |
| <b>Resume</b>                                      | yes  | yes                                | yes  | yes                                | yes   | no  |
| <b>Cross-<br/>platform</b>                         | yes  | yes                                | yes  | yes                                | yes *<br>Cygwin   | yes * Perl  |
| <b>Easy<br/>setup<br/>and<br/>mainten<br/>ance</b> | FASTCopy/<br>yes<br>other<br>products/??                         | no demo                            | no demo  | client/mod<br>erate<br>server/?    | yes   |   |
| <b>Technic<br/>al<br/>Support</b>                  | good.<br>phone, web,<br>email.<br>Excellent<br>documentati<br>on | -                                  | late<br>response<br>to request<br>for<br>pricing | web                                | little  | none  |
| <b>Cost</b>  | \$1000/work<br>station   | ????<br>server/<br>\$300<br>client | \$6500/w<br>orkstatio<br>n                       | \$???<br>server/<br>free client    | none*<br>developm<br>ent cost to<br>add<br>automatic<br>retry | development<br>cost to fix<br>security,<br>reliability,<br>add resume |

The Lab for High Energy Astrophysics Data Transfer System (DTS - <http://heasarc.gsfc.nasa.gov/dts/>) is listed in the last column. This has been favored by the GSSC, but was not favored by the MOC because of the following:

- The data notification occurs via e-mail. This was criticized because there is no guarantee that e-mail will be delivered. (We will come back to the e-mail issues later.)
- It was thought that DTS did not have the capability of picking up where it left off during a file transfer. This capability (called "split") was not mentioned in the older version of the DTS user's guide. It is now included in version 6.1 available from the web site [http://heasarc.gsfc.nasa.gov/dts/protect/dts\\_ug.pdf](http://heasarc.gsfc.nasa.gov/dts/protect/dts_ug.pdf)

- It was also not known that there is e-mail support ([dtshelp@athena.gsfc.nasa.gov](mailto:dtshelp@athena.gsfc.nasa.gov)) because the DTS web site had listed the wrong e-mail address for outside users - [dtshelp@olegacy.gsfc.nasa.gov](mailto:dtshelp@olegacy.gsfc.nasa.gov), which is not reachable from outside Goddard.

The MOC then installed FASTcopy and checked that some of the features advertised by Softlink (FASTcopy's manufacturer). They found that indeed it seemed robust in its file transfer mechanism and liked the flexibility I configuration.

## **2.2. GSSC trade study**

The MOC's preference for FASTcopy led the GSSC to make a detailed evaluation of FASTcopy to see if it could best meet the requirements. We installed an evaluation copy on two GSSC computers, tried transfers between them and also with the MOC. We read the FASTcopy documentation, and corresponded with Softlink support to solve installation problems and get clarification on features and operational details.

We are also taking a closer look at DTS. At this time we do not have DTS installed, but we have the advantage of sharing offices with the two DTS developers and can get very detailed information about DTS. DTS is now being installed for testing at the GSSC.

Our first task was to understand the features and operations available from FASTcopy and DTS. We have done a side-by-side comparison of DTS and Fastcopy (see our section titled: Features comparison: DTS and FASTcopy). The upshot is that the features in DTS and FASTcopy are quite comparable. There are some areas where the FASTcopy implementation and proposed usage of certain features were worrisome to us. We detail these in the next section.

### **2.2.1. Issues of Concern for FASTcopy and DTS**

This section details the issues that were raised in evaluating FASTcopy. We should note, that as proposed, the MOC wants to use FASTcopy in push mode, where the file sender pushes files to a receiver. We are unhappy with FASTcopy in this mode for reasons detailed below. We note that probably a workaround could be found for all of these concerns about FASTcopy, but our view is that this would require more software development to create something that comes already written in DTS. These concerns go back to the last three items listed in the above list of requirements. Overall we felt that satisfying the data accountability requirements would be easier with the way DTS is implemented.

These issues of concern are listed in the following table:

| <b>Issues of concern for using FASTcopy over DTS</b> |  |  |
|--|--|--|
| <b>Issue</b>   | <b>DTS</b>   | <b>FASTcopy</b>  |
| Pushing data/disk space check                        | Push list of files, which initiates immediate pull – transfer process can check disk availability first. | Can push files even when no disk space is available.   |
| Software reliability on Linux                        | Uses standard perl   | Installation script had several problems. – No support for Fedora Linux yet.   |
| Checkpointing  | Effective checkpoint transfer – files is split by sender, pieces put back together on receiver           | Checkpointing by similar method  |
| Post transfer processing                             | Locally controlled by receiver   | In push mode – sender must know receiver command to be executed  |
| Receipt path   | Local control of directory to put data   | In push mode sender must specify directory to place transferred files.   |
| Logging  | Provides log of transfer   | Logging functions did not seem to work on Linux. “Context files” appear only on transfer initiator host, and disappear after transfer is complete. |
| Round trip data verification/data accountability     | List sent in e-mail triggers files to transfer – Ack sent to sender.                                     | Must be done by hand – sender would push a list of files – list needs to be checked against received data.   |
| e-mail server security concerns                      | Uses e-mail server (workarounds possible with fetchmail or ssl tunneling)                                | No e-mail usage.   |
| Port access  | Need mail port waiver  | Need socket port waiver  |
| Long term usage (13 years)                           | Have access to source and can maintain ourselves if necessary.   | Proprietary code – depends on existence of company.  |
| Cost   | Free   | \$2k for two machines (prime and backup) and ~\$300(?) /year support   |

### 2.2.2. Other issues with FASTcopy:

- Installation/Support problems with FASTcopy – Does not run on support current versions of Linux (RH EE, or Fedora) and required several rewrites of the UNIX install script to get it to install on a supported Linux platform).
- Logging problems with FASTcopy (Could not get server to log anything – created an empty log file).

### 2.2.3. MOC Concerns for DTS

The biggest concerns about using DTS center on e-mail issues:

- The main weakness we see with DTS is that it does rely on e-mail for data availability notification, which is not guaranteed. We don't feel this is a serious problem, since only the e-mail could get lost and not data, and another e-mail can always be sent. We are also comforted by the fact that DTS has been in operation for years with XMM and not a single piece of data has been lost.
- The use of sendmail and having open port 25. There are ways around this
  - Use of ssl tunneling for all communication investigated by Rob Preece and student – this allows one to close port 25 open, and also to avoid sendmail (they recommend postfix).
  - Fetchmail can grab mail from an outside computer so again port 25 can be blocked and sendmail need not be run. (Fetchmail is used this way in the RXTE Science Operations Facility.)

We think that these solutions will address the concerns raised by the MOC.

### 2.2.4. Features comparison: DTS and FASTcopy

We did some investigation into how various features work and have summarized our findings in the table below.

| Feature                   | DTS   | FASTCopy  |
|---------------------------|---|---|
| Acknowledgement to sender | DTS 'get' mode notifies sender an acknowledgement of successful transfer. This allows clean up at sender.                       | Receiver can send an acknowledgement via a post-transfer command execution to the sender.             |
| Clean up                  | Allows for cleaning up of staging area via config file or command line option.  | Allows for cleaning up of staging area via local post-transfer command.                               |
| Command execution         | Can activate external scripts via configuration files.  | Can activate external scripts (pre and post, local and remote commands).                              |
| Command execution         | To execute a post-transfer command, sender and receiver have to agree on tags that identify specific operations to be executed. | Sender also needs to know about the remote-post-transfer command if receiver wants it to be executed. |

|                         |  |  |
|-------------------------|--|--|
| Communication           | Uses Email and SFTP (OpenSSH) servers.   | Uses sockets for communication.  |
| Compression             | Allows compression/decompression on the fly.   | Allows compression/decompression on the fly.   |
| Data transfer mechanism | DTS 'send' compiles and sends a list of files (with stats – size and checksum on each file) to receiver in an Email. Email stays in receiver's local DTS mailbox until DTS 'get' is started. 'get' can be started interactively or via a cronjob. Note that DTS creates this list of file automatically. | Sender can execute a pre-transfer command to notify receiver of new data availability to allow it to 'pull' data; or initiate a 'push' with fcopy command and transfer data in receiver's pre-specified directory structure.   |
| Error reporting         | Log files are generated on both sender and receiver sites. Logs transactions and errors and forwards to human DTS operator at receiver site. Errors are sent to the site that DTS determines most likely to be interested in the message.  | Allows generation of report at sender site. No way for receiver to know if something went wrong during the transfer. A post-transfer command execution can remedy this somewhat.   |
| Logging                 | Record progress messages during a transfer on both sides.  | Record progress messages during a transfer to a context file at sender site.   |
| Portability             | Works on Solaris, Digital UNIX and Linux.  | Works on Windows, Linux and Solaris.   |
| Recovery                | Option to split and recombine files. (This allows the resume from point of failure functionality)  | Internally splits and recombines files to perform transfer. (This allows the resume from point of failure functionality)   |
| Recovery                | Number of transfer attempts can be set via configuration file. If all tries fail, the operation is aborted.  | Batch run automatically starts the recovery process until it succeeds. One can set a number of attempts and/or time between each attempt. If all tries fail, the operation is aborted.   |
| Recovery                | Recovery transfers all unmatched/un-transferred files. One can choose to split the files before transfer. This will allow recovery almost at the point of last failure. The split files automatically re-constitute themselves.  | Capable of transferring only the differences between files at a block level over the network for efficient replication.<br>Recovery starts at the point of failure in a file. If the file has changed, one can use 'force' qualifier to force recovery from the point of last change. 'force' is only available in interactive mode. |

|                        |  |  |
|------------------------|--|--|
| Recovery               | Number of transfer attempts can be set via configuration file. If all tries fail, the operation is aborted.  | Manual recovery for interactive operations.  |
| Recovery               | Can use split and send option for large files to accomplish the same thing as FASTCopy.  | Hold and resume capability for very large transfers.   |
| Remote directory tree  | Can reconstitute directory structure.  | Can recursively copy directory structures and recreate them on the target machine.   |
| Remote directory tree  | DTS puts all transferred files into one directory in the staging area at receiver site. File list contains original directory structure, and it is up to the local site to move the files back to their proper directory tree (if desired).<br>A script to do this is available. | A 'push' makes it necessary for sender to know the directory structure of the receiver. Or we can put all files in same directory and follow what DTS does to relocate the transferred files.  |
| Retransmission request | Since the user "pulls" data, receiver can request a retransmission, if needed.   | Retransmission request – receiver seems to have no control over this.  |
| Run mode               | Can be run interactively or in batch.  | Can be run interactively or in batch.  |
| Validation             | Verifies transferred file validity by checking checksum and byte size checks.  | Provides various levels of verifications – <ul style="list-style-type: none"> <li>• Network – 'on the fly' verification of each transmitted packet</li> <li>• File – post-transfer comparison of the whole source and destination files</li> <li>• Protocol – synchronized write of target files to disk</li> <li>• Recovery – checking source and destination files for mismatch during recovery</li> </ul> |

### 2.2.5. Other options:

The LAT has raised the possibility of using bbcp (<http://www.ihep.ac.cn/~chep01/paper/7-018.pdf>). While this program has certain features (optimizing TCP windows sizes to maximize bandwidth), it does not provide many of the bookkeeping and external features provided by DTS and FASTcopy. We do feel that bbcp is worth considering as a replacement for sftp in DTS, however, and intend to study the issue in more detail.